AFRL-VA-WP-TP-2004-305

COMMUNICATION REQUIREMENTS IN THE COOPERATIVE CONTROL OF WIDE AREA SEARCH MUNITIONS USING ITERATIVE NETWORK FLOW

Jason W. Mitchell, Steven J. Rasmussen, and Andrew G. Sparks

FEBRUARY 2004

Approved for public release; distribution is unlimited.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

QREPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS**.

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
February 2004	Conference Paper	

4. TITLE AND SUBTITLE
COMMUNICATION REQUIREMENTS IN THE COOPERATIVE CONTROL OF WIDE AREA SEARCH MUNITIONS USING ITERATIVE NETWORK FLOW

5a. CONTRACT NUMBER
F33615-01-D-3105

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62201F

6. AUTHOR(S)
Jason W. Mitchell and Steven J. Rasmussen (General Dynamics Advanced Information Systems)

Andrew G. Sparks (AFRL/VACA)

5d. PROJECT NUMBER
A05V

5e. TASK NUMBER
0F

5f. WORK UNIT NUMBER
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
General Dynamics Advanced Information Systems
1919 S. Eads St.
Arlington, VA 22202

(AFRL/VACA)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/VACA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-VA-WP-TP-2004-305

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Conference Proceedings for the Conference on Cooperative Control and Optimization, Destin, FL, November 19-21, 2003. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. This report contains color.

14. ABSTRACT ***(Maximum 200 Words)***

Communication requirements are considered for the cooperative control of wide area search munitions where resource allocation is performed by an iterative network flow. We briefly outline both the single and iterative network flow assignment algorithms and their communication requirements. Then, using the abstracted communication framework recently incorporated into AFRL's MultiUAV simulation package, a model is constructed to investigate the peak and average data rates occurring in a sequence of vehicle-target scenarios using an iterative network flow for task allocation, implemented as a redundant, centralized optimization, that assumes perfect communication.

15. SUBJECT TERMS
cooperative control, uninhabited aerial vehicles, information flow, communication requirements, data rates, MultiUAV

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Andrew G. Sparks
Unclassified	Unclassified	Unclassified	SAR	22	19b. TELEPHONE NUMBER *(Include Area Code)* (937) 255-8682

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Chapter 1

COMMUNICATION REQUIREMENTS IN THE COOPERATIVE CONTROL OF WIDE AREA SEARCH MUNITIONS USING ITERATIVE NETWORK FLOW*

Jason W. Mitchell*, Steven J. Rasmussen†
General Dynamics Advanced Information Systems
Wright-Patterson AFB, OH 45433-7531
{Jason.Mitchell,Steve.Rasmussen}@wpafb.af.mil

Andrew G. Sparks‡
Air Force Research Laboratory
Wright-Patterson AFB, OH 45433-7531
Andrew.Sparks@wpafb.af.mil

Abstract Communication requirements are considered for the cooperative control of wide area search munitions where resource allocation is performed by an iterative network flow. We briefly outline both the single and iterative network flow assignment algorithms and their communication requirements. Then, using the abstracted communication framework recently incorporated into AFRL's MultiUAV simulation package, a model is constructed to investigate the peak and average data rates occurring in a sequence of vehicle-target scenarios using an iterative network flow for task allocation, implemented as a redundant, centralized optimization, that assumes perfect communication.

Keywords: cooperative control, uninhabited aerial vehicles, information flow, communication requirements, data rates, MultiUAV

*This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.
*Aerospace Scientist
†Senior Aerospace Engineer
‡Aerospace Scientist

1. Introduction

Coordination and cooperation between uninhabited aerial vehicles (UAV) has the potential to significantly improve their effectiveness in many situations. For the typical tasks that these vehicles must perform, i.e. detection, classification, attack, and verification, explicit vehicle cooperation may be required to meet specific objectives. Thus, the ability to communicate information between vehicles becomes mission essential and provides an opportunity to enhance overall capability.

While vehicle communications may provide the opportunity to enhance performance, it is likely not without cost. Frequently, control algorithms are designed without regard to their associated communication needs or effects. For the control system designer, such treatment is undertaken to reduce algorithmic complexity and obtain a manageable result. Consequently, it becomes necessary to quantify the communicated data driving the control algorithms ex post facto. As an example of this design strategy, consider several methods that have been previously studied to produce near-optimal single task assignments [10, 6], and more recently, the near-optimal assignment of a sequence of tasks using an iterative network flow model [11]. In these cases, the amount of information necessary to drive these cooperative control algorithms was not considered.

In this work, communication requirements are considered for the cooperative control of uninhabited aerial vehicles with resource allocation performed by an iterative network flow. In the following, we briefly outline the single and iterative network flow assignment algorithms and their communication requirements. Then, we briefly describe the MultiUAV simulation package [7, 9], and the framework recently incorporated to model vehicle-to-vehicle communication. Using this framework, a model is constructed to investigate the peak and average data rates occurring in a sequence of vehicle-target scenarios using an iterative network flow for task allocation, implemented as a redundant, centralized optimization, that assumes perfect communication.

2. Background

We begin with a short description of a typical MultiUAV simulation scenario and a brief outline of the network flow task allocation models.

The current configuration of MultiUAV simulates, but is not limited to, autonomous wide area search munitions (WASM), which are small UAVs powered by a turbojet engine with sufficient fuel to fly for a short period of time. They are deployed in groups from larger aircraft flying at higher

altitudes. Individually, they are capable of searching for, recognizing, attacking, and verifying targets.

2.1. Scenario

We begin with a set of N vehicles, deployed simultaneously, each with a life span of approximately thirty (30) minutes, that are indexed by $i \in \mathbb{Z}[1, N]$. Targets that may be found by searching fall into known classes according to the value or score associated with their destruction. These targets are indexed by j as they are found, thus we find $j \in \mathbb{Z}[1, M]$ with V_j as the value of target j. The individual vehicles assume no precise a priori information available about the total number of targets or their initial locations. This information can only be obtained by the vehicles searching for and finding potential targets via Automatic Target Recognition (ATR) methodologies. The ATR process is modeled using a system that provides a probability that the target has been correctly classified. The probability of a successful classification is based on the viewing angle of the vehicle relative to the target, Rasmussen et al. [9]. For this exercise, the possibility of incorrect identification is not modeled, however targets are not attacked unless a 90% probability of correct identification is achieved. Further details of the ATR methodology can be found in Chandler and Pachter [2], with a detailed discussion available in Chandler and Pachter [1]. Once successfully classified as a target, the attack vehicle is selected. Upon reaching the selected target, the vehicle releases its munition and is subsequently declared an unavailable asset, i.e. attack is a terminal task for WASM. Finally, the selected target must be verified as destroyed to complete the target specific task chain.

Throughout the simulation, at each target state change or task failure, a resource allocation algorithm is executed to compute task assignments. The resulting assignment is sub-optimal. Fortunately, Rasmussen et al. [8] has shown that these assignments are typically near-optimal in an average sense.

2.2. Task Allocation: Network Optimization Model

The weapon system allocation is treated as follows: individual vehicles are discrete supplies of single units, executing tasks corresponding to flows on arcs through the network, with the ultimate disposition of the vehicles representing the demand. Thus, the flows are zero (0) or one (1). We assume that each vehicle operates independently, and makes decisions when new information is received. These decisions are determined by the solution of the network optimization model. The receipt

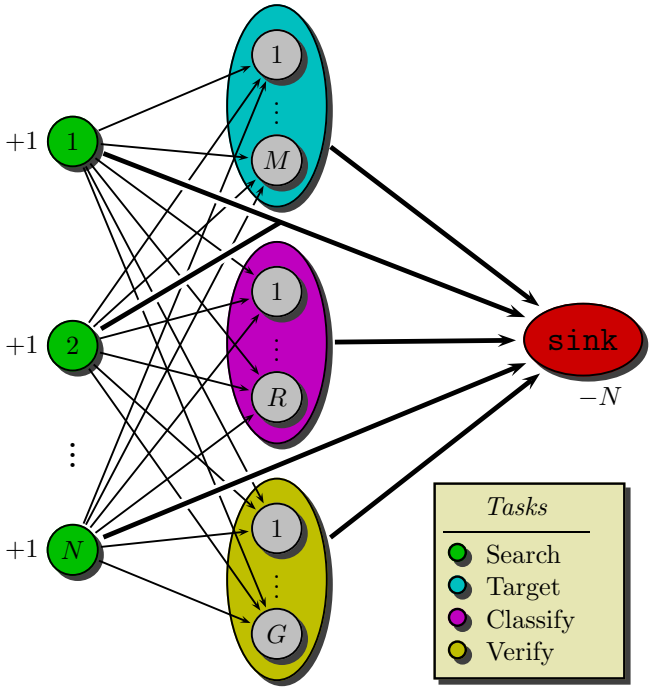

Figure 1.1: Network flow diagram.

of new target information triggers the formulation and solving of a fresh optimization problem that reflects current conditions, thus achieving feedback action. At any point in time, the database on-board each vehicle contains a *target* set, consisting of indices, types and locations for targets that have been classified above the probability threshold. There is also a *speculative* set, consisting of indices, types and locations for potential targets that have been detected, but are classified below the probability threshold and thus require further inspection before striking.

The network flow model, seen in Figure 1.1, is demand driven. The `sink` node at the right exerts a demand-pull of N units, causing the nodes on the left to flow through the network. In the middle layer, the top M nodes represent all of the successfully classified targets, and thus are ready to be attacked. An arc exists from a specific vehicle node to a target node if and only if it is a feasible vehicle/target pair. At a minimum, the feasibility requirement would mean that there is sufficient fuel remaining to strike the target if so tasked. Other feasibility conditions could also be considered, e.g. heterogeneous weapons or sensing platforms, poor look-angles. The center R nodes of the middle

layer represent potential targets that have been detected, but do not meet the minimum classification probability. We call them *speculatives*. The minimum feasibility requirement to connect a vehicle/speculative pair is sufficient fuel for the vehicle to deploy its sensor to elevate the classification probability. The lower-tier G nodes model alternatives for verification of targets that have been struck. Finally, each node in the vehicle set on the left has a direct arc to the far right node labelled `sink`, modeling the option of continuing to search. The capacities on the arcs from the target and speculative sets are fixed at one (1). From the integrality property, flow values are constrained to be either zero (0) or one (1). Each unit of flow along an arc has a benefit which is an expected future value. The optimal solution maximizes total value. For a more detailed discussion, including the issue of the *benefit calculation*, see Schumacher et al. [11].

2.2.1 Single Pass Network Flow. Single task assignment in MultiUAV is formulated as the capacitated transshipment problem (CTP) [10]. Due to the special structure of the problem, there will always be an optimal solution that is all integer [6]. Thus, solutions to this problem pose a small computational burden, making it feasible for implementation on the processors likely to be available on inexpensive wide area search munitions.

2.2.2 Iterative Network Flow. Due to the integrality property, it is not normally possible to simultaneously assign multiple vehicles to a single target, or multiple targets to a single vehicle. However, using the network assignment iteratively, *tours* of multiple assignments can be determined [11]. This is done by solving the initial assignment problem once, and only finalizing the assignment with the shortest estimated arrival time. The assignment problem can then be updated assuming that assignment is performed, updating target and vehicle states, and running the assignment again. This iteration can be repeated until all of the vehicles have been assigned terminal tasks, or until all of the target assignments have been fully distributed. The target assignments are complete when classification, attack, and verification tasks have been assigned for all known targets. Assignments must be recomputed if a new target is found or a munition fails to complete an assigned task.

2.3. Information Requirements

The implementation of the task allocation algorithms outlined above requires communication of information between vehicles. As with several previous studies where MultiUAV was used to investigate optimal task

allocation, we assume perfect and error-free access to information about vehicle and target states. From many perspectives, these assumptions are clearly unrealistic, particularly when considering physical communication and processing constraints. However, to determine the requirements of a physically realizable system, we must also understand what information is necessary and the quantity needed to drive the algorithms under ideal conditions.

Since both algorithms discussed here make use of network flow, the necessary information is common between them. The overarching optimization problem can be characterized as both centralized and redundant, i.e. each vehicle computes its own network flow. Momentarily disregarding communication issues, the problem, in general, requires a synchronized database of target and vehicle state information. With this, each vehicle computes the benefits for the arcs in the network, and solves the optimization problem to maximize the total benefit. From Mitchell et al. [5], the MultiUAV network flow implementation requires the following communicated information: ATR data; target and vehicle positions; target, vehicle, and task status; and vehicle trajectory waypoints.

Having identified the information necessary, we can begin to consider the volume of information communicated between vehicles. To do this, we turn to the MultiUAV simulation package.

3. Simulation Framework

The MultiUAV simulation package [9] is capable of simulating multiple uninhabited aerospace vehicles which cooperate to accomplish a predefined mission. The purpose of the package is to provide a simulation environment that researchers can use to implement and analyze cooperative control algorithms. The simulation is built using a hierarchical decomposition where inter-vehicle communication is explicitly modeled. The package includes plotting tools and provides links to external programs for post-processing analysis. Each of the vehicle simulations include six-degree-of-freedom dynamics and embedded flight software (EFS). The EFS consists of a collection of *managers* or *agents* that control situational awareness and responses of the vehicles. In addition, the vehicle model includes an autopilot that provides waypoint navigation capability. In its original form, MultiUAV [7] could simulate a maximum of eight (8) vehicles and ten (10) targets, however recent work eases the previous burden of extending these limits. The EFS managers implement the cooperative control algorithms, including the iteratively applied CTP algorithm previously discussed. The individual

managers contained within the vehicles include: Tactical Maneuvering, Sensor, Target, Cooperation, Route, and Weapons. At the top level, these managers are coded as SIMULINK models, with supporting code written in both MATLAB script and C++.

3.1. Communication Model

The communication simulation used in this work is very similar to that used in Mitchell et al. [5]. However, in this instance, communication is not delayed, so that the messages,[1] generated by the simulated vehicle communication at each major model update, arrive in the *in-box* of a given vehicle at the completion of the current update, and are available for use at the next major update. At the present time, the major model occurs at 10 Hz. This fairly course grained update is necessary to maintain a reasonable run-time for individual scenarios to complete, in a larger Monte-Carlo sense, on a desktop/personal computer. The minor model update, which controls the vehicle dynamics and other underlying subsystems, is scheduled at 100 Hz.

As a consequence of the model update rates, we define the *data rate* necessary at a given major model step as the total size of the messages collected, in bits, divided by the duration of the model update, yielding a rate in bits/s. This simplistic definition is a result of the elementary requirement that each vehicle must have access to all the currently generated messages by the next major update in order to function. Currently, all message data is represented in MATLAB using double-precision floating-point numbers, and in the computation of data rate, the message overhead is not considered, only the message payload. In a physical communication implementation there would be considerably more overhead, including redundancy, error correction, encryption, etc. Thus, retaining double-precision in the ideal communication model remains a reasonable indicator of real-world data rates, particularly since we are interested only in an initial estimate and perhaps a relative comparison of communication necessary in executing various scenarios. Furthermore, a *broadcast* communication model is implicitly assumed, so that generated messages are counted only once. While not specifically targeted to address a particular physical implementation, such a model encompasses the typical view that the communications are time-division multiplexed.

[1] The use of *message* here refers to the information format dictated by the MultiUAV package, rather than to messages related to a specific communication system model or protocol.

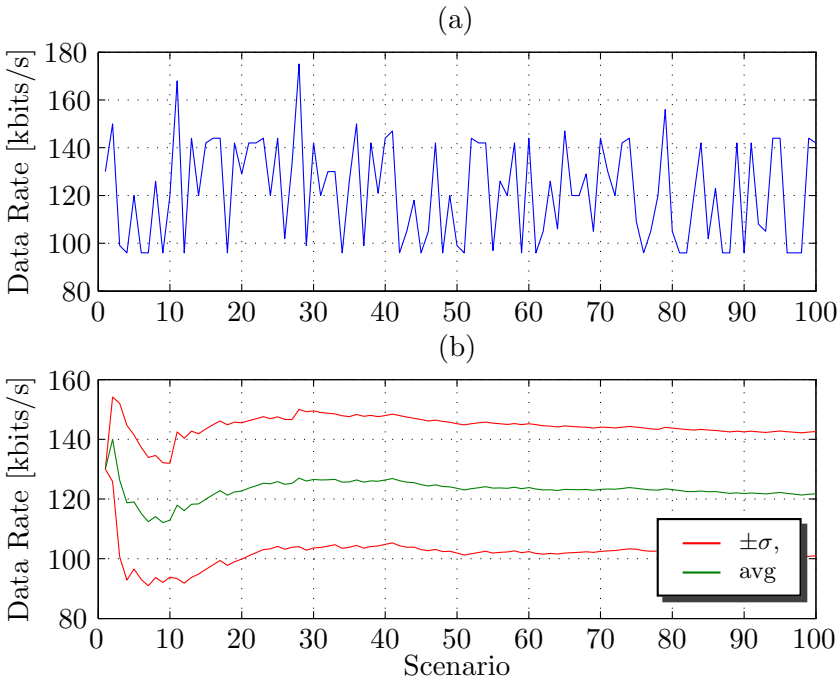

Figure 1.2: Maximum data rate (a) and cumulative average (b) over 100 simulations.

4. Simulation

In this work, we investigate the communication data rate requirements for the cooperative control of wide area search munitions using a iterative network flow of depth three (3). To study this, a Monte-Carlo approach is taken, consisting of one hundred (100) individual simulations, each with a maximum mission time of $t_f = 200\,\text{s}$.

Individual scenarios are composed of eight (8) vehicles with four (4) targets distributed over an area of approximately $16\,\text{mi}^2$. The vehicle properties are: constant velocity of 370 ft/s or approximately mach 0.33, constant altitude of 675 ft, minimum turn radius of 2000 ft, and fuel for a maximum of 30 min of search operation. Since search is not the focus of this study, vehicles begin in a line formation, and initially follow a preprogrammed *zamboni race* search pattern. The targets are uniformly distributed throughout the domain and oriented with uniformly random pose-angles.

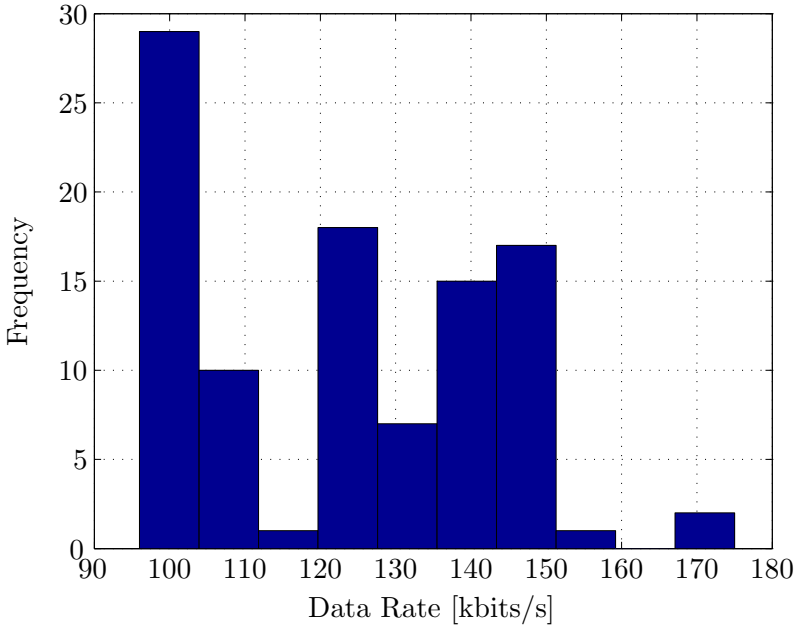

Figure 1.3: Maximum data rate frequency distribution of 100 simulations.

5. Results

As a simple measure to convince ourselves that the Monte-Carlo data collected has sufficient statistical weight, we plot the maximum data rate for all 100 simulations, and compute the cumulative average, seen in Figure 1.2. Surprisingly, we see that there is considerable variation in the maximum data rate. Fortunately, in terms of statistical weight, we see that the average maximum data rate is within 0.2 % of the final cumulative average after just 50 simulations. This is not surprising based on previous work in performing Monte-Carlo simulation with MultiUAV [8, 4, 5]. From the distribution of maximum data rates seen in Figure 1.3, it appears that the largest number fall between 120 –150 kbit/s. Most of the remaining data is distributed at a lower maximum data rate centered around 105 kbit/s. The single remaining maximum rate is centered at 170 kbit/s.

From this information, we see that, for the given model update resolution and iterative network flow cooperative control algorithm, a significant data rate is required for operation. This obviously ignores consideration of any hardware or software to mitigate communication delay

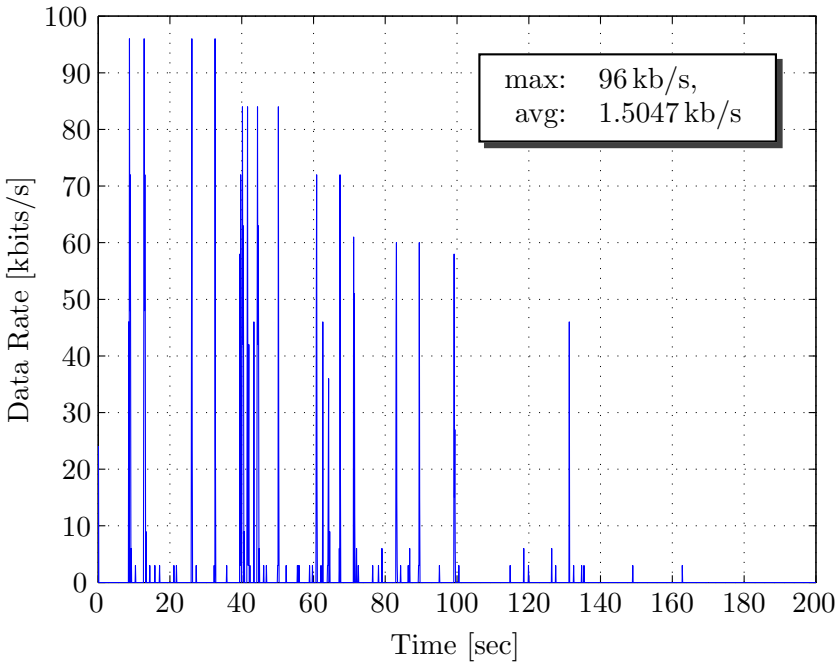

Figure 1.4: Communication history: smallest maximum data rate.

effects or insure information integrity that are likely to be included in a physical implementation. Nevertheless, by disregarding the actual magnitude of the maximum data rate, and considering only a relative measure between scenarios, we find that the largest data rate necessary is nearly twice the smallest maximum data rate.

Rather than attempt to analyze each individual simulation run, it is more interesting to compare the scenarios representing the smallest, average, and largest maximum data rates: 96 kbit/s, 120 kbit/s, and 175 kbit/s, respectively. The corresponding communication data rate histories can be seen in Figures 1.4–1.6, respectively.

For the smallest maximum data rate, seen in Figure 1.4, the peaks are well spaced, and decrease as targets are destroyed. Based on the distribution of data rates, Figure 1.3, this appears to be the less frequent of two typical operational modes. For the average maximum data rate, given by Figure 1.5, we find the more typical communication situation. For this scenario, the rate peaks are much more closely spaced, and do not always decrease as targets are destroyed due to the spike at $t \approx 35$ s. There is also considerable communication activity for $t \in [80, 100]$ s.

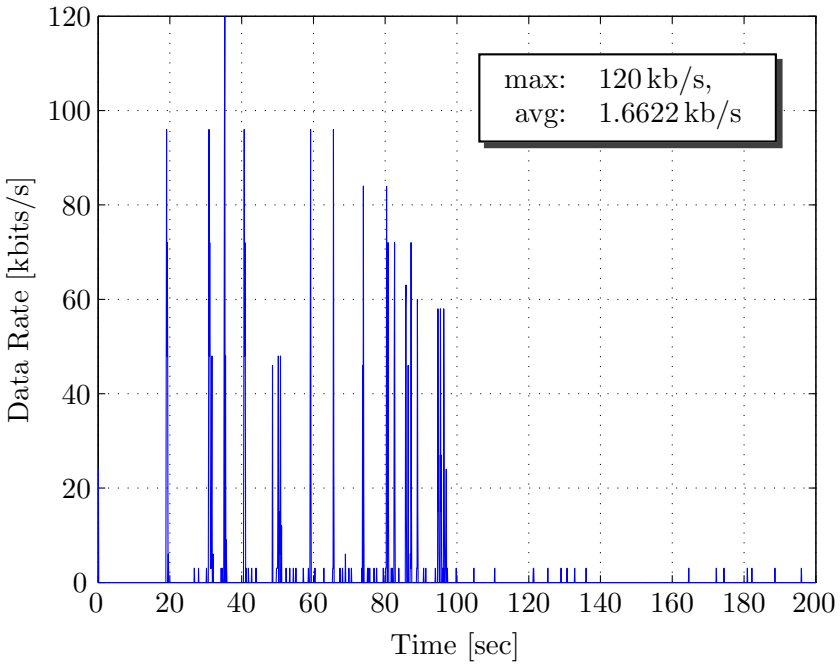

Figure 1.5: Communication history: average maximum data rate.

Lastly, for the largest data rate, found in Figure 1.6, the magnitude of the largest peak is nearly twice that of the other rate peaks occurring. Given this information, it is instructive to study the vehicle trajectories for the corresponding scenarios. These trajectories appear in Figures 1.7–1.9, where vehicles are identified by a `typewriter` style, e.g. 2, and targets are identified by an *italics* style, e.g. *2*, so that they may be more easily distinguished.

The vehicle trajectories for the smallest maximum data rate are found in Figure 1.7. The trajectory traces are relatively simple, particularly since targets appear in two clusters: *1*,*3* and *2*,*4*. For the communication burst around $t \approx 40\,\mathrm{s}$, we find that a target classification has failed, requiring further classification. For the average data rate seen in Figure 1.8, the vehicle trajectories are much more complex, with considerable looping and backtracking. Again, we notice that targets appear in two clusters: *1* and *2*,*3*,*4*. However, the second cluster contains three targets. The spike at $t \approx 35\,\mathrm{s}$ results from a failed classification attempt, while the end communication bursts are a result of the three-target cluster. Lastly, for the largest data rate, given by Figure 1.9, the vehicle tra-

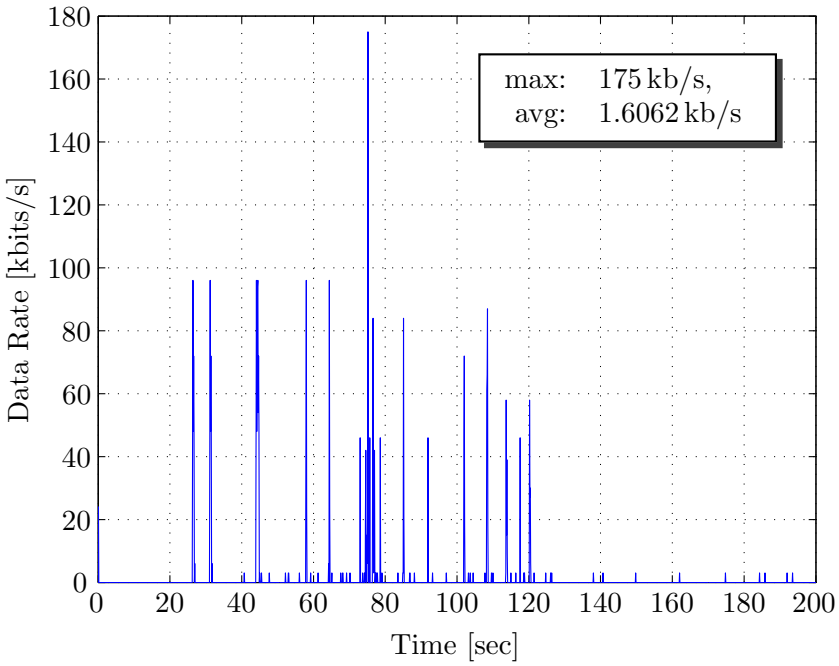

Figure 1.6: Communication history: largest maximum data rate.

jectories are extremely convoluted. The target clustering is similar to the smallest maximum data rate case, but with the target clusters placed closer together. This explains the communication burst at $t \approx 75\,\mathrm{s}$. In addition, at the time of the largest spike, a number of failures occur. At $t = 74.3\,\mathrm{s}$, classification of target *2* fails. Then, at $t = 74.9\,\mathrm{s}$, two classifications, viz. targets *3* and *4*, fail simultaneously. At $t = 75.3\,\mathrm{s}$, target *4* is successfully classified. Following this, at $t = 76.3\,\mathrm{s}$, target *2* is discovered, then viewed a second time and successfully classified, at $t = 76.6\,\mathrm{s}$. The second classification resulted in a task being completed by a vehicle not assigned to that task, producing an additional task failure. Overall, this particular scenario appears to be a quite pathological case of task sequencing.

In summary, the Monte-Carlo data indicates that there were two primary operational communication modes. In a relative comparison sense, the mode corresponding to the smaller maximum data rate, centered at 105 kbit/s, represents scenarios with a lower incidence of task failure, or lower target density. As the number of task failures or target density increases, the maximum data rate increases to accommodate the addi-

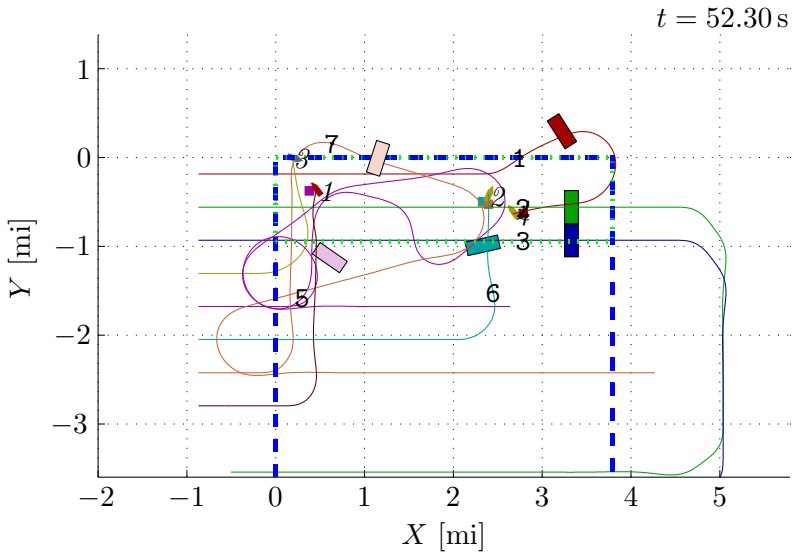

Figure 1.7: Vehicle Trajectories: smallest maximum data rate.

tional information necessary to make more frequent decisions; ranging between 120 kbit/s and 150 kbit/s. For the remaining case, we see that pathological task sequencing composed of both simultaneous task failures and simultaneous events generates the largest maximum data rate of 175 kbit/s.

6. Conclusions

In this work, the communication requirements were considered for the cooperative control of wide area search munitions. Using the MultiUAV simulation package, a model was constructed to investigate the peak and average data rates occurring in a sequence of vehicle-target scenarios using an iterative network flow for task allocation, which was implemented as a redundant, centralized optimization. This model assumed perfect vehicle-to-vehicle communication. The data rate was defined to be the amount of data communicated during a major model update divided by the major update duration, were each element was represented by a double-precision floating-point value.

The communication data rate indicated that when a mission scenario suffered setbacks, such as failed tasks, event accumulation bursts, or

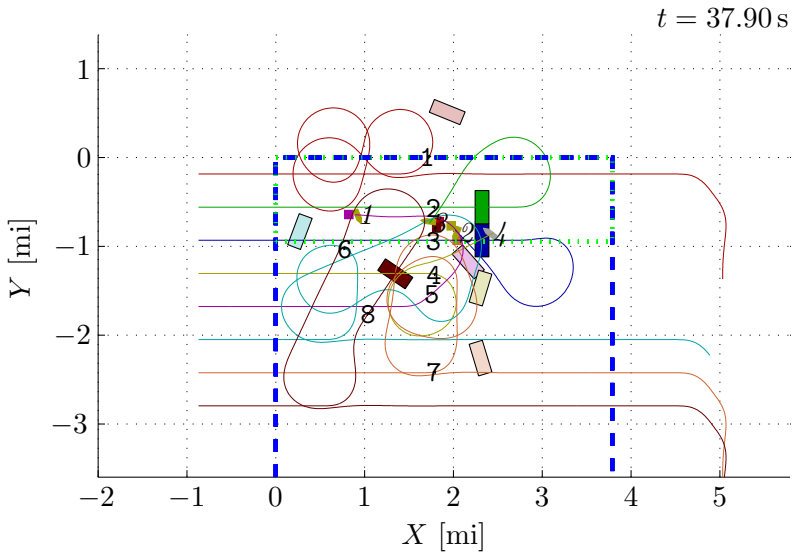

Figure 1.8: Vehicle Trajectories: average maximum data rate.

other difficulty, mission performance suffered, even with perfect communication. This information clearly represented a relative measure of mission health, even during execution.

Having observed that the structure of the communication data rate history correlated well with the likelihood that a particular scenario suffered some difficulty, we hope this quantification of cooperation may be used as a measure to help maintain a desired level of coordination. Such a measure could be used, for example, to ensure the graceful degradation of mission performance in the presence of constrained information flow between vehicles.

Regarding the actual magnitudes of the maximum data rates, these should not be taken as exact requirements or measures, particularly because no specific communication protocol or hardware implementation has been defined. Rather, the magnitudes should be seen to represent traditional engineering estimates that say more in their relative significance than individual significance. With that said, these values do indicate the amount of raw data necessary to drive the cooperative control algorithms, allowing for comparisons between individual implementations of an algorithm.

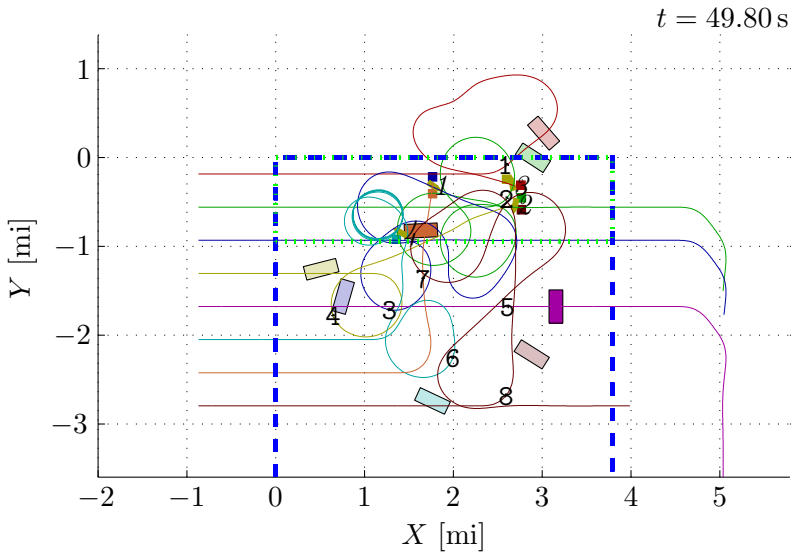

Figure 1.9: Vehicle Trajectories: largest maximum data rate.

Acknowledgments

A portion of this work was performed while the first named author held a National Research Council Research Associateship Award at the Air Force Research Laboratory (AFRL) in the Air Vehicles Directorate Control Theory Optimization Branch (VACA) located at the Wright-Patterson Air Force Base.

References

[1] Phillip R. Chandler and Meir N. Pachter. Hierarchical control of autonomous teams. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2001.

[2] Phillip R. Chandler and Meir N. Pachter. UAV cooperative classification. In *Workshop on Cooperative Control and Optimization*. Kluwer Academic Publishers, 2001.

[3] L.R. Ford, Jr. and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.

[4] Jason W. Mitchell, C. Schumacher, Phillip R. Chandler, and Steven J. Rasmussen. Communication delays in the cooperative

control of wide area search munitions via iterative network flow. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2003.

[5] Jason W. Mitchell and Andrew G. Sparks. Communication issues in the cooperative control of unmanned aerial vehicles. In *Proceedings of the Forty-First Annual Allerton Conference on Communication, Control, & Computing*, 2003.

[6] Kendall E. Nygard, Philip R. Chandler, and M. Pachter. Dynamic network flow optimization models for air vehicle resource allocation. In *Proceedings of the Automatic Control Conference*, 2001.

[7] Steven J. Rasmussen and Philip R. Chandler. MultiUAV: A multiple UAV simulation for investigation of cooperative control. In *Proceedings of the Winter Simulation Conference*, 2002.

[8] Steven J. Rasmussen, Phillip R. Chandler, Jason W. Mitchell, C. Schumacher, and Andrew G. Sparks. Optimal vs. heuristic assignment of cooperative autonomous unmanned air vehicles. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2003.

[9] Steven J. Rasmussen, Jason W. Mitchell, Chris Schulz, C. Schumacher, and Phillip R. Chandler. A multiple UAV simulation for researchers. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, 2003.

[10] C. Schumacher, Philip R. Chandler, and Steven J. Rasmussen. Task allocation for wide area search munitions via network flow optimization. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2001.

[11] C. Schumacher, Philip R. Chandler, and Steven J. Rasmussen. Task allocation for wide area search munitions via iterative network flow optimization. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2002.